Osnovi digitalne elektronike IR

vežba 2

Odsek za elektroniku

Elektrotehnički fakultet,
Univerzitet u Beogradu

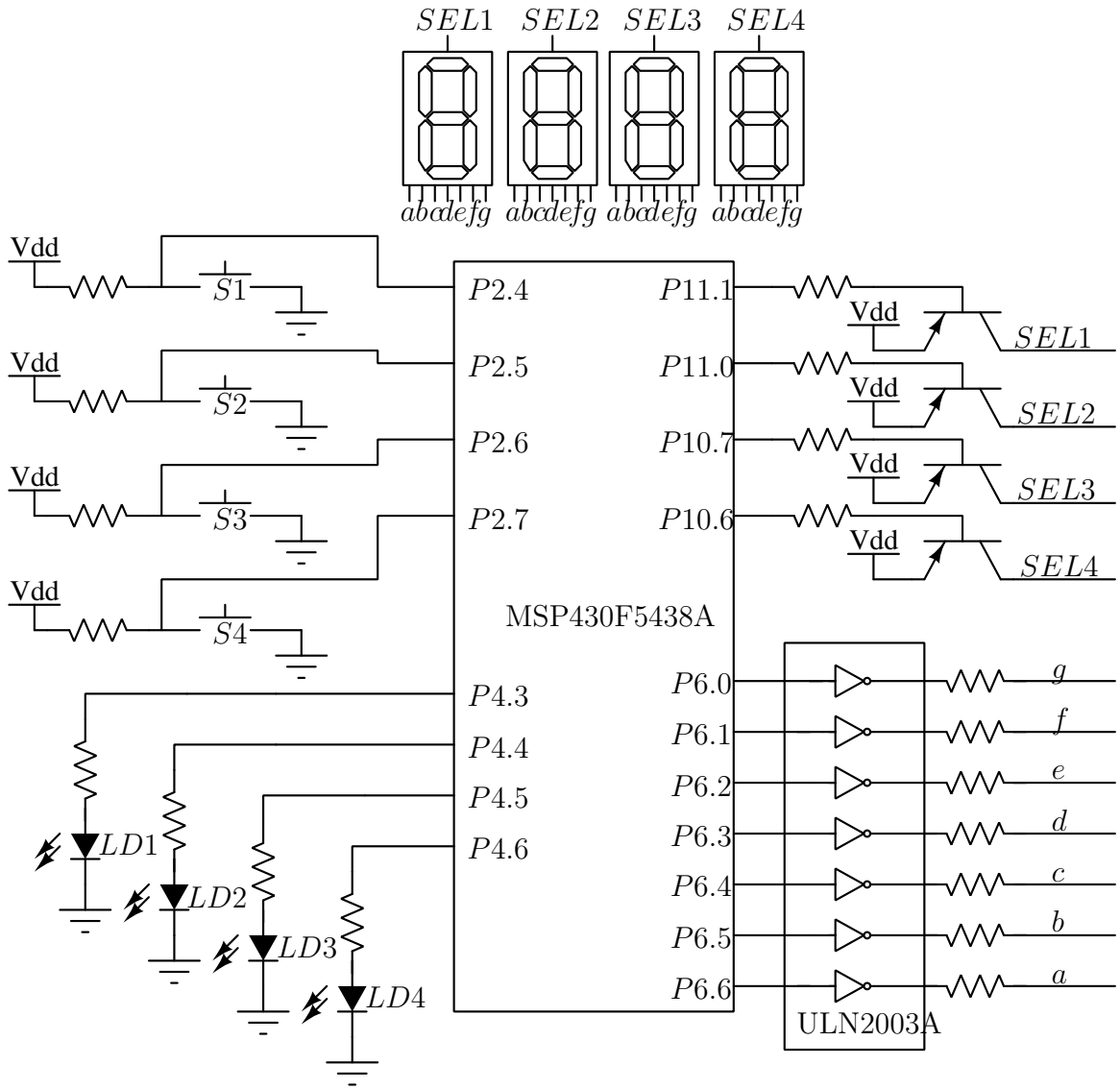
2018/2019

Pregled

1. Razvojno orkuženje
   - Hardversko razvojno okruženje
     - Portovi
   - Softversko razvojno okruženje
     - *Include* fajlovi
     - Linkovanje
     - Kompajliranje
     - Debagovanje
2. Primeri
   - Rad sa diodama
   - Rad sa tasterima
   - Prekidi
   - Sedmosegmentni displej

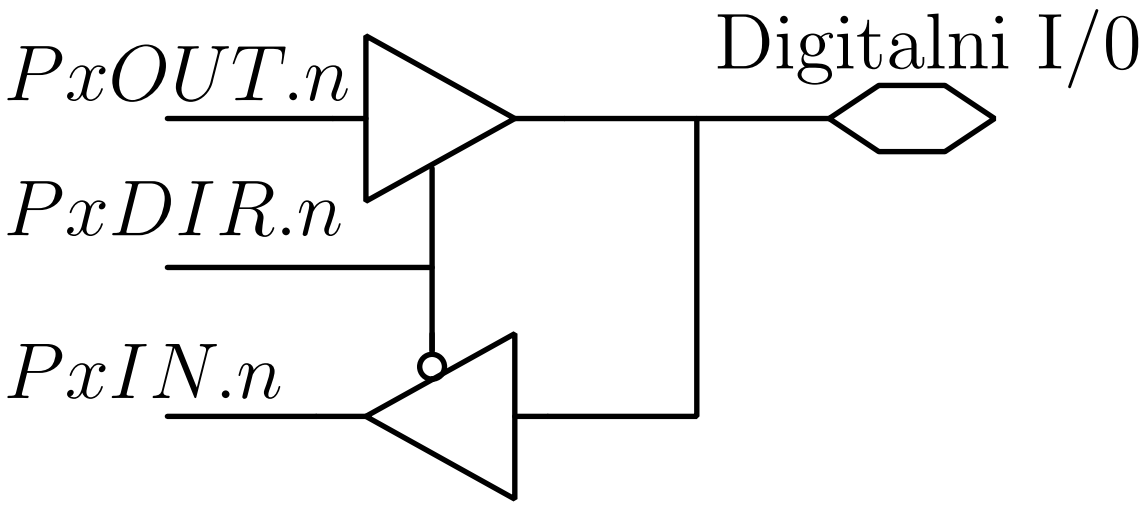
Pregled

Opis hardvera



Digitalni portovi

Svaki port se kontroliše sa četiri registra



Registri za kontrolu portova 1/2

PxIN - ulazni registar ($x = 1 - 11$)

- očitani bit = 0 - na ulazu je nizak logički nivo
- očitani bit = 1 - na ulazu je visok logički nivo

PxOUT - izlazni registar ($x = 1 - 11$)

- upisani bit = 0 - na izlazu je nizak logički nivo
- upisani bit = 1 - na izlazu je visok logički nivo

PxDIR - registar selekcije ulaza ili izlaza ($x = 1 - 11$)

- upisani bit = 0 - pin je ulazni
- upisani bit = 1 - pin je izlazni

PxSEL - registar selekcije alternativnih funkcija ($x = 1 - 11$)

- upisani bit = 0 - ulazno/izlazna funkcija pina

Registri za kontrolu portova 2/2 (Registri kontrole prekida)

PxIFG - registar flegova ($x = 1, 2$)

- očitani bit = 0 - na pinu se nije dogodila promena koja izaziva prekid
- očitani bit = 1 - na pinu se dogodila promena koja izaziva prekid

PxES - registar selekcije ivice ($x = 1, 2$)

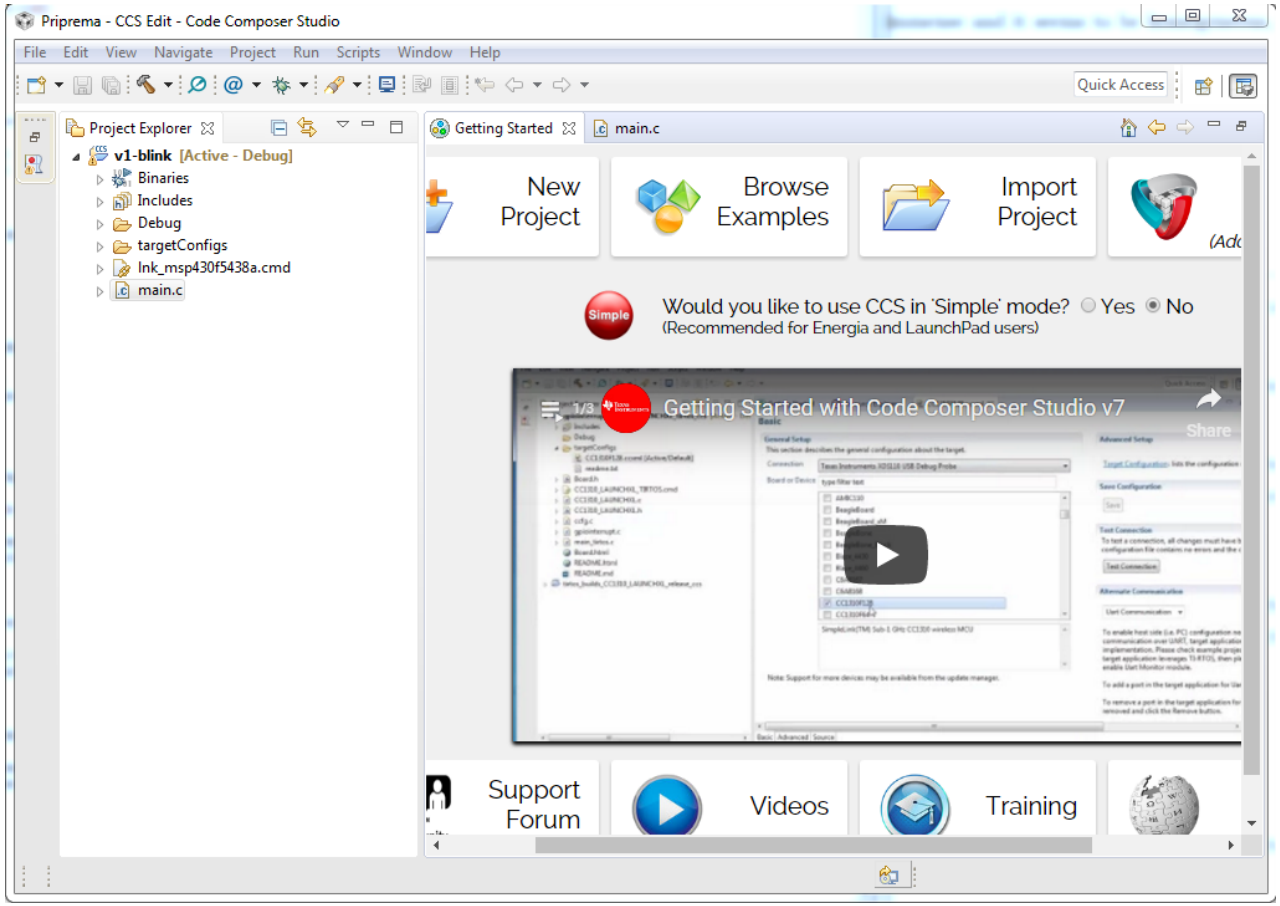
- upisani bit = 0 - prekid se događa na rastuću ivicu signala na pinu
- upisani bit = 1 - prekid se događa na opadajuću ivicu signala na pinu

PxIE - registar dozvole prekida ($x = 1, 2$)

- upisani bit = 0 - prekid nije dozvoljen za promenu stanja na pinu
- upisani bit = 1 - prekid je dozvoljen za promenu stanja na pinu

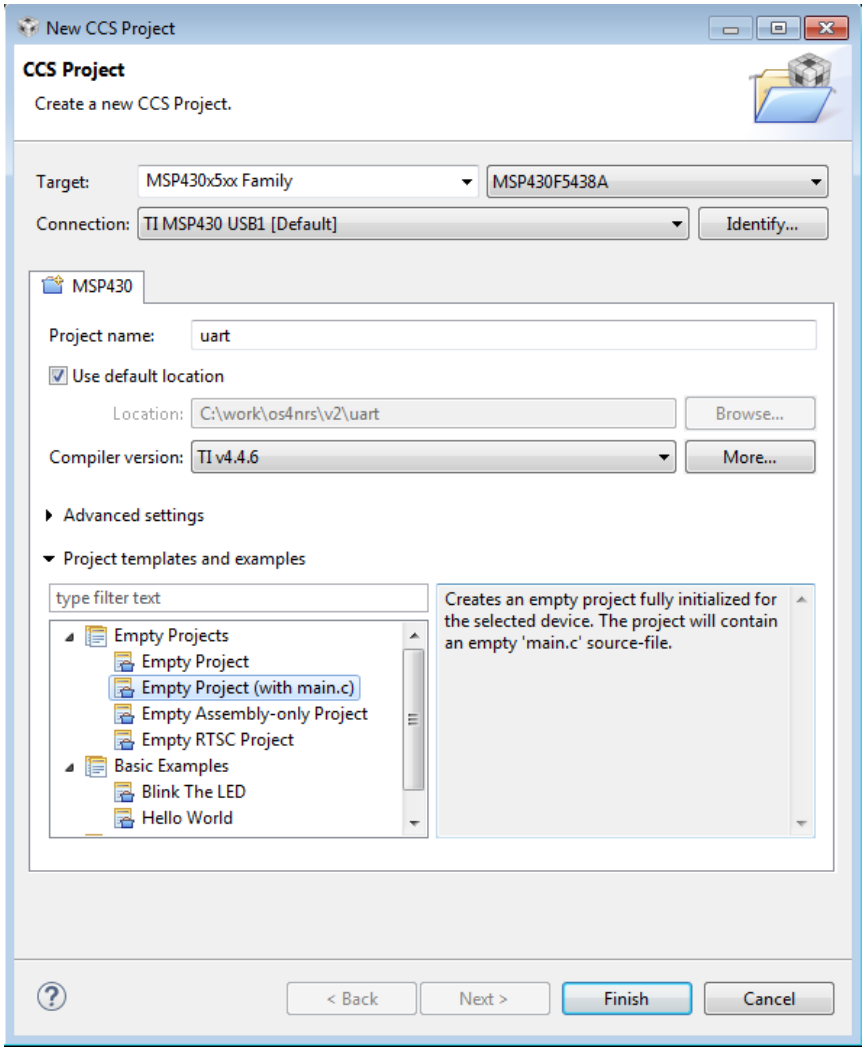
Code Composer Studio V7

Preuzimanje softvera moguće sa linka
`http://www.ti.com/tool/CCSTUDIO`



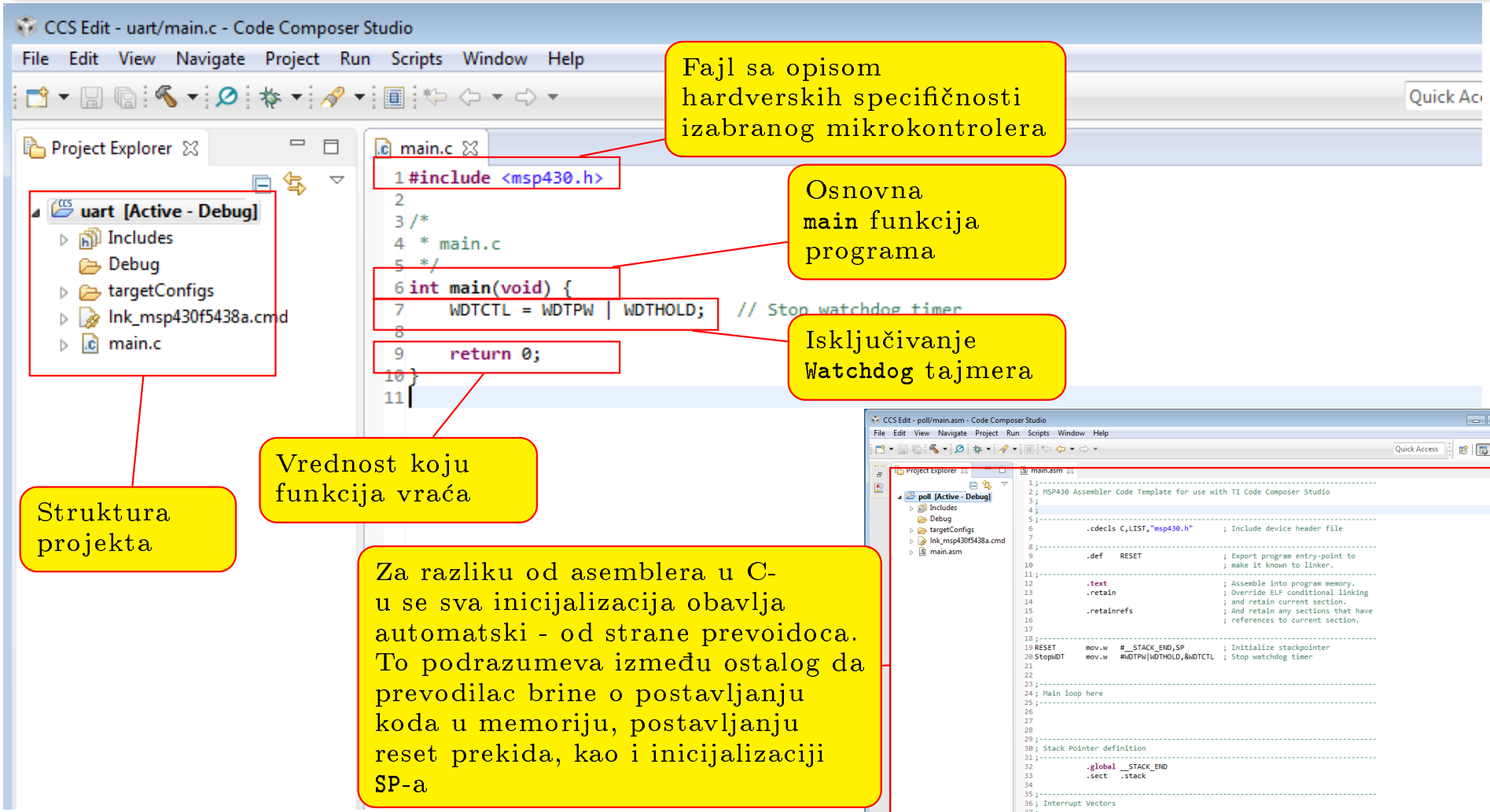
Kreiranje projekta

S obzirom da je tražena implementacija u `C`-u, u dijalogu kreiranja projekta bira se C-ovski template



Osnovni fajl

Predefinisani template za C-ovski projekat prikazan je na slici



Include fajl

Korišćenje pojmova definisanih u *include* fajlu olakšava kodiranje ali i kasniji prelaz sa jednog na drugi mikrokontroler

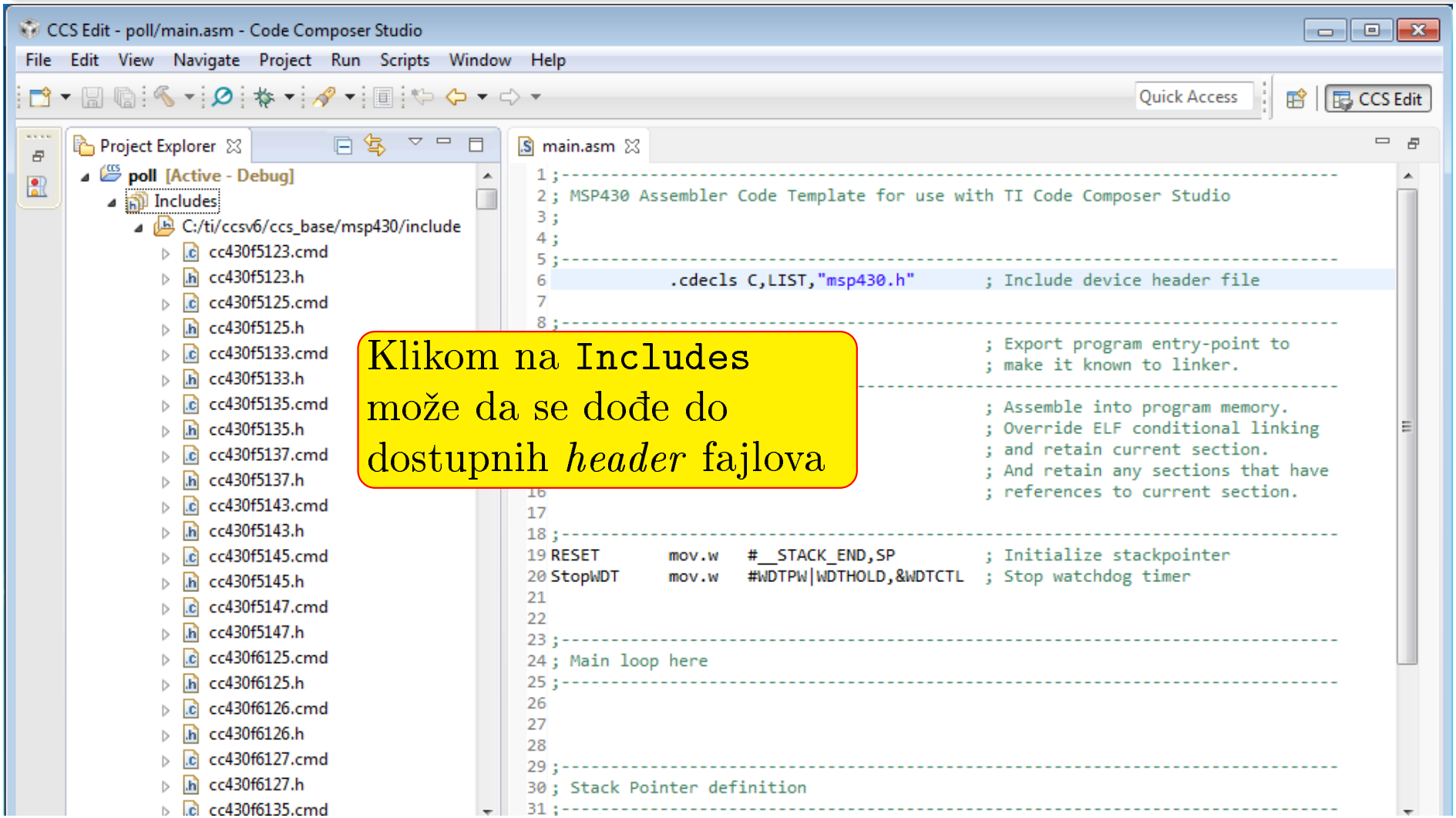
Makroima se simbolička imena registara povezuju sa fizičkim adresama na magistrali

Često korišćenim konstantama zadaju se simbolička imena

```
mov.w   #WDTPW|WDTHOLD,&WDTCTL  ; Stop
#define WDTPW                   (0x5A00)
```

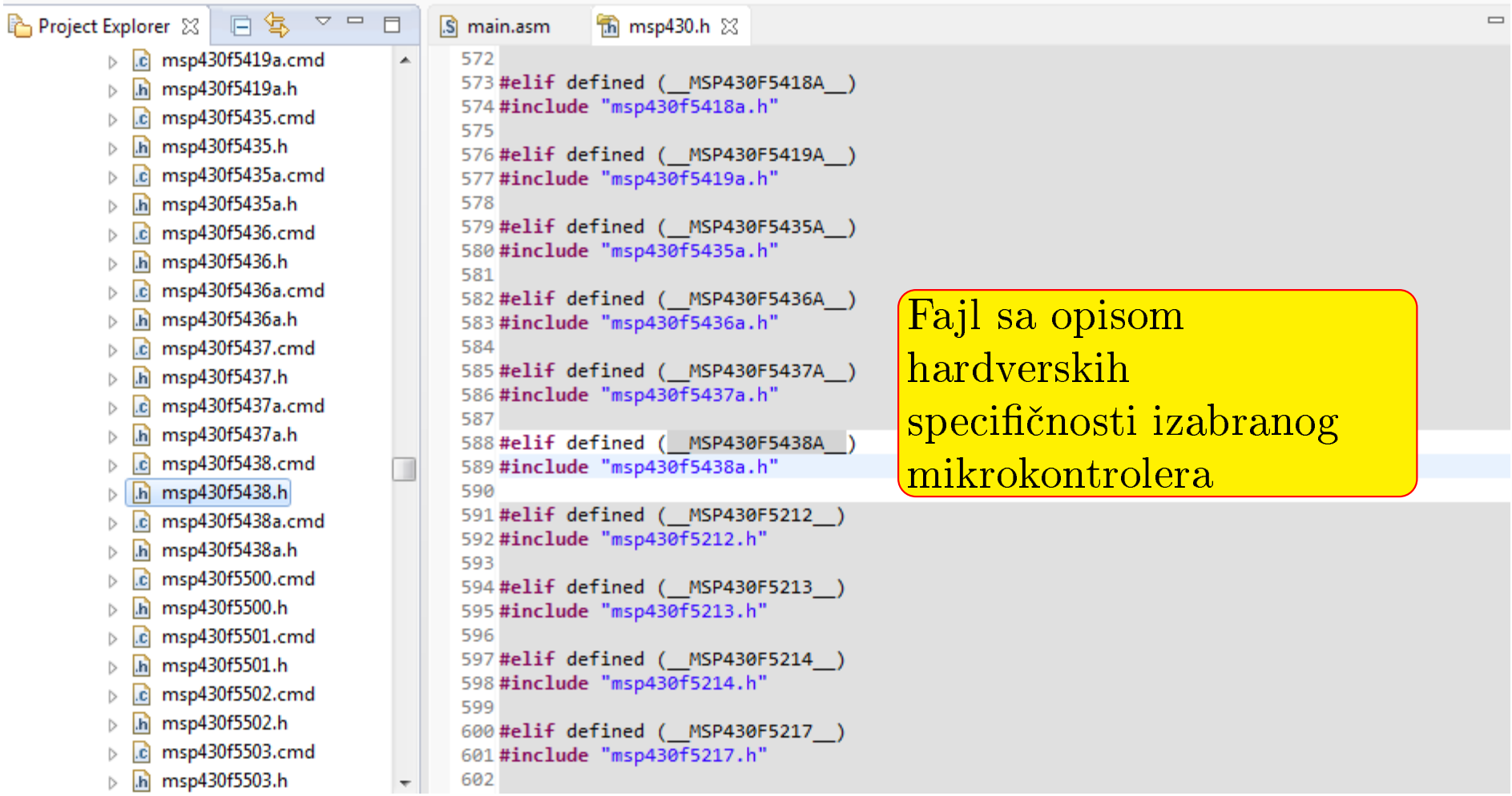
Hardverske specifičnosti 1/2

`msp430.h` je opšti *include* fajl za celu MSP430 familiju



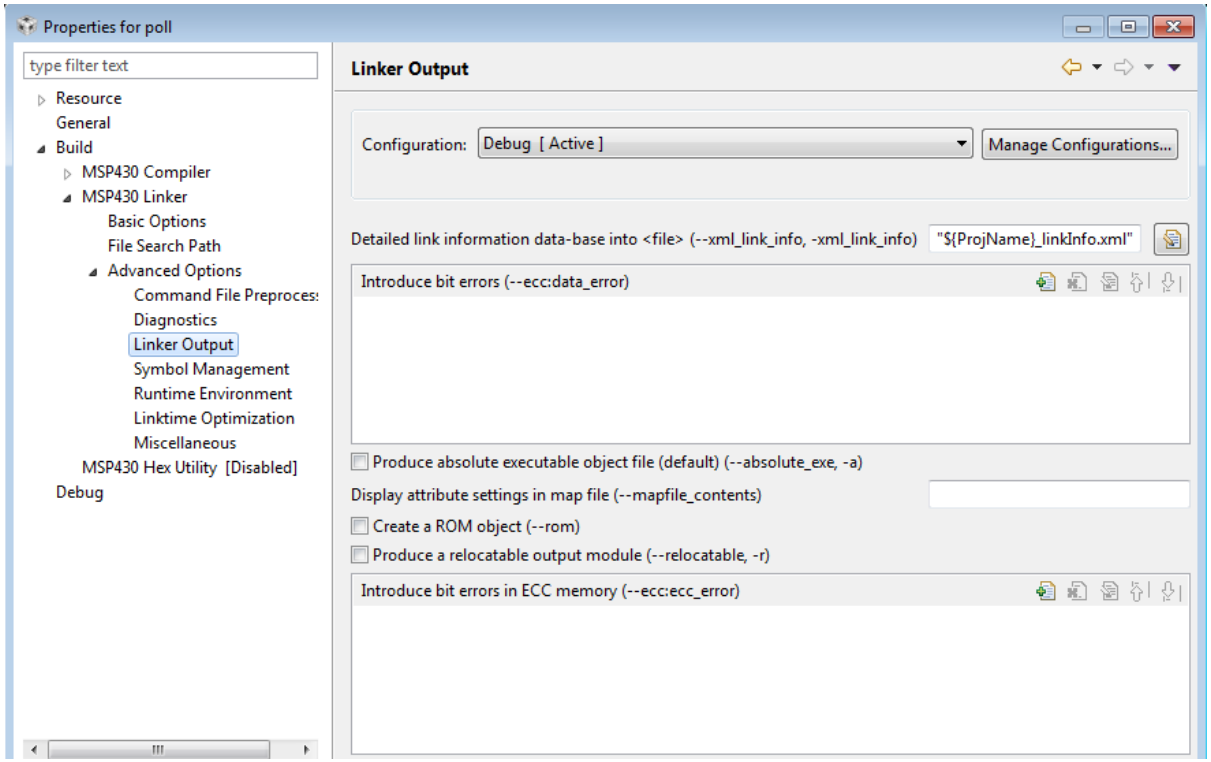
Hardverske specifičnosti 2/2

Unutar fajla `msp430.h` definisanim makroima se uključuje fajl za izabrani mikrokontroler



Linkovanje

Linkovanje je proces u kome se mašinski kod iz različitih asemblerskih fajlova prevodi u jedan jedinstven fajl koji se može spustiti u memoriju mikrokontrolera.



Najvažnije uputstva za linker se nalaze u *linker command file*-u

lnk_msp430f5438a.cmd

```
133 /* Specify the sections allocation into
134 /*************************************************************************/
135
136 SECTIONS
137 {
138     .bss        : {} > RAM                  /* Global & static vars
139     .data       : {} > RAM                  /* Global & static vars
140     .TI.noinit  : {} > RAM                  /* For #pragma noinit
141     .sysmem     : {} > RAM                  /* Dynamic memory allocation
142     .stack      : {} > RAM (HIGH)           /* Software system stack
143
144 #ifndef __LARGE_DATA_MODEL__
145     .text       : {}>> FLASH                /* Code
146 #else
147     .text       : {}>> FLASH2 | FLASH       /* Code
148 #endif
149     .text:_isr  : {} > FLASH                /* ISR Code space
150     .cinit      : {} > FLASH                /* Initialization tables
151 #ifndef __LARGE_DATA_MODEL__
152     .const      : {} > FLASH                /* Constant data
153 #else
154     .const      : {} > FLASH | FLASH2       /* Constant data
155 #endif
156     .cio        : {} > RAM                  /* C I/O Buffer
157
158     .pinit      : {} > FLASH                /* C++ Constructor tables
159     .init_array : {} > FLASH                /* C++ Constructor tables          */
160     .mspabi.exidx : {} > FLASH              /* C++ Constructor tables          */
161     .mspabi.extab : {} > FLASH              /* C++ Constructor tables          */
162
163     .infoA      : {} > INFOA            /* MSP430 INFO FLASH Memory segments */
164     .infoB      : {} > INFOB
165     .infoC      : {} > INFOC
166     .infoD      : {} > INFOD
167
168     /* MSP430 Interrupt vectors          */
169     .int00      : {}                > INT00
170     .int01      : {}                > INT01
```

Segmenti mogu da se preklapaju

.bss je tip segmenta u koji se smeštaju neinicijalizovani podaci (RAM memorija)

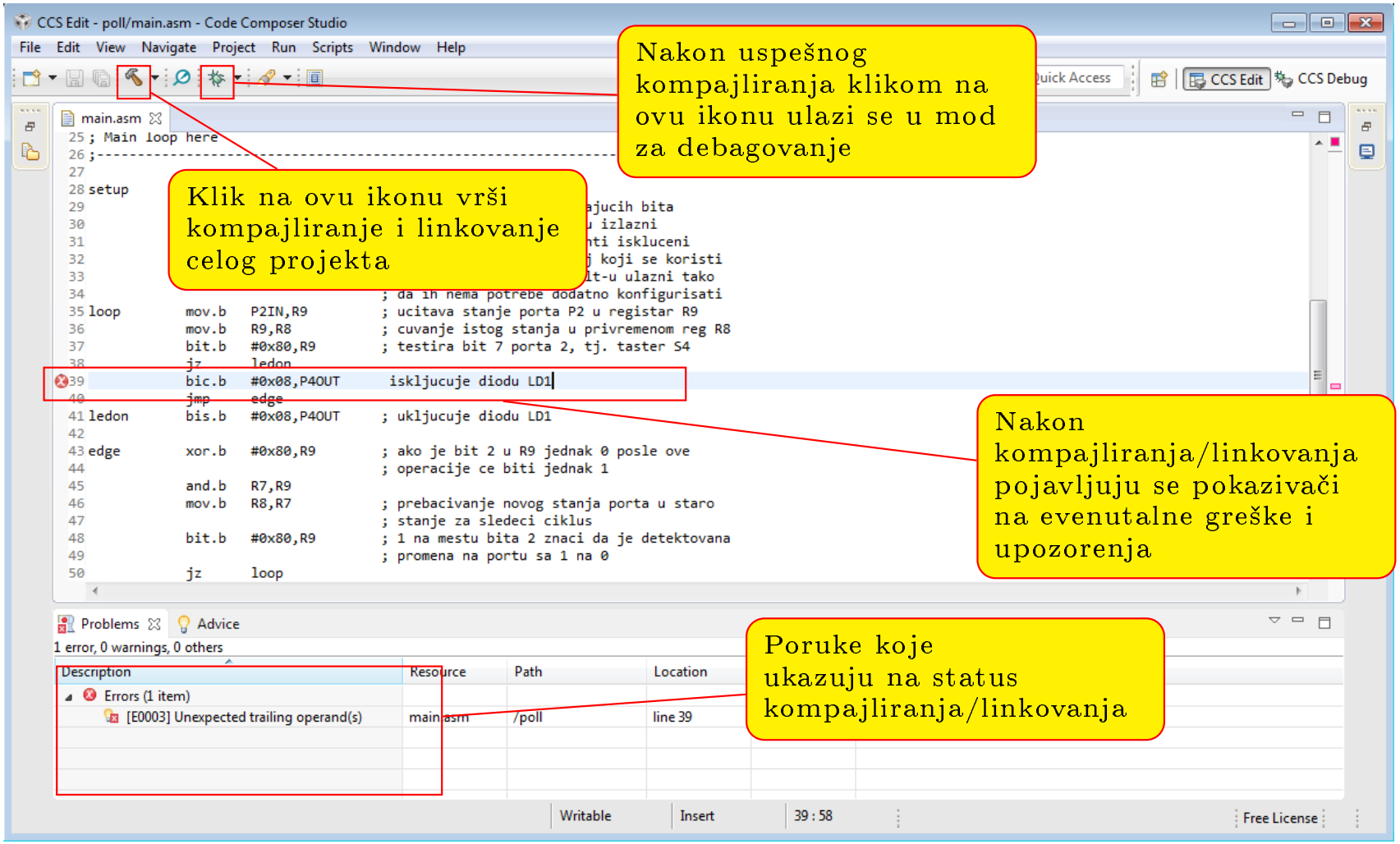
.data je tip segmenta u koji se smeštaju inicijalizovani podaci (RAM memorija)

.text je tip segmenta u koji se smeštaju instrukcije (*FLASH* memorija)

.const je tip segmenta u koji se smeštaju konstantni podaci (*FLASH* memorija)

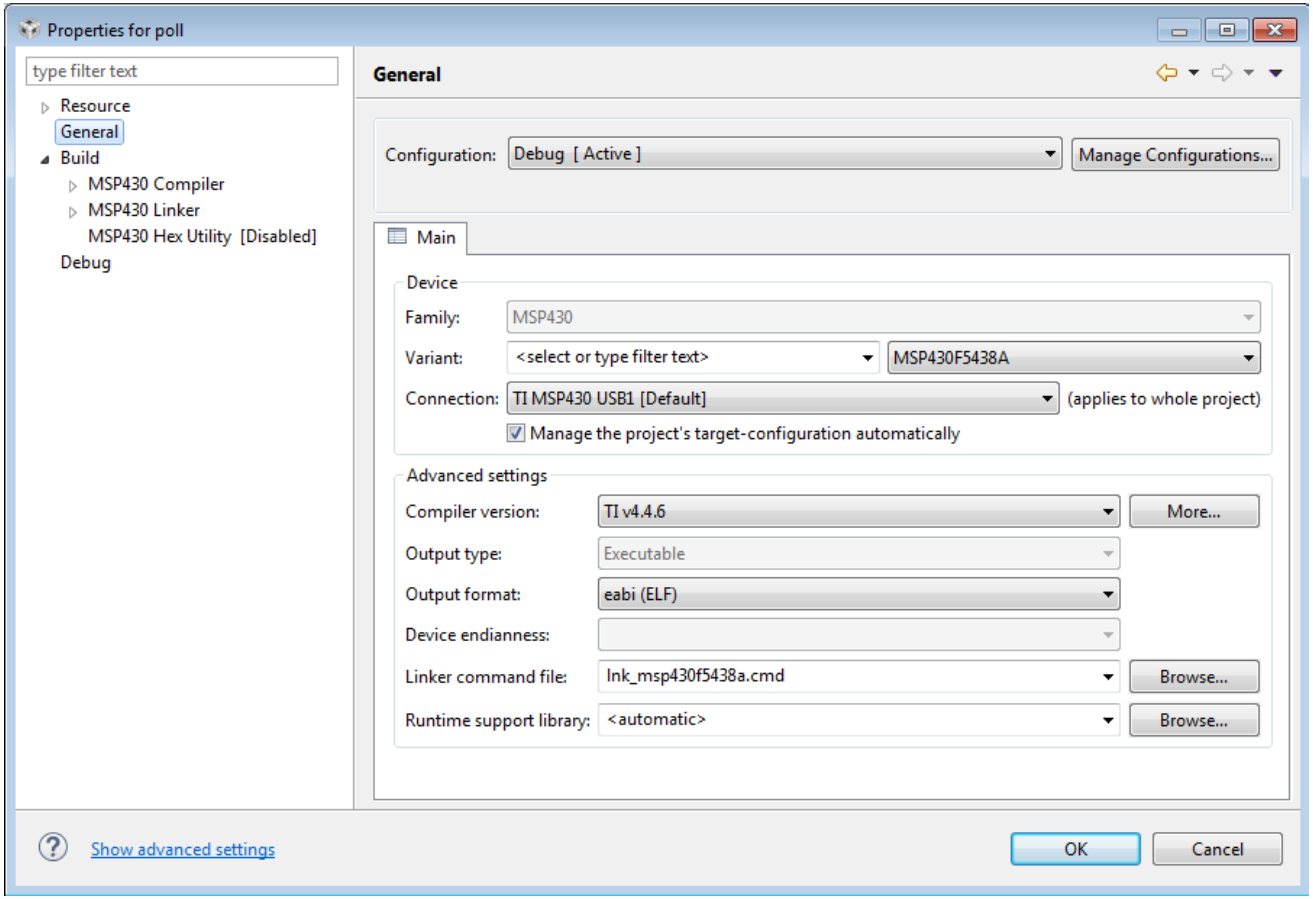
Kompajliranje

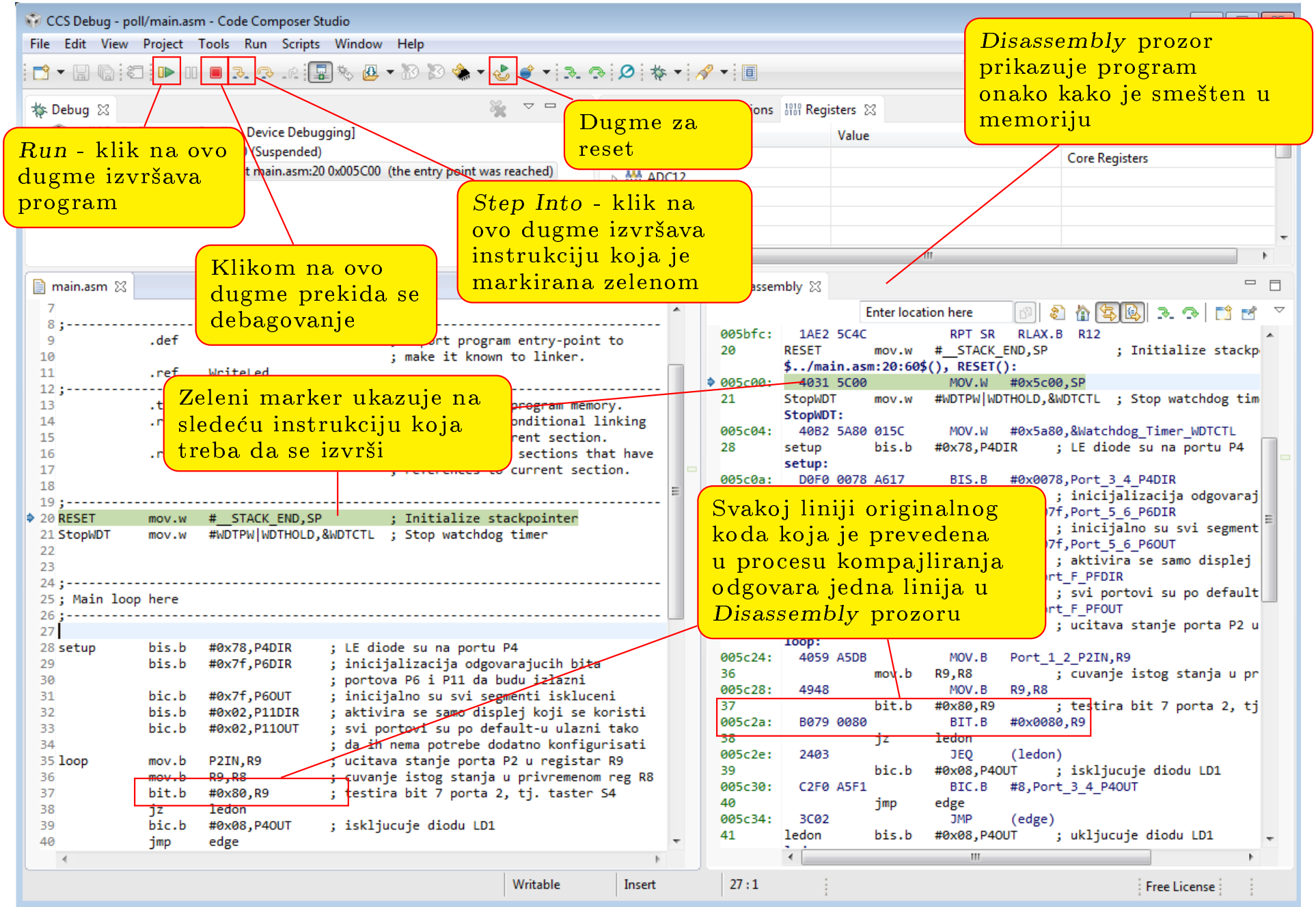
Komanda `BUILD` vrši kompajliranje i linkovanje projekta

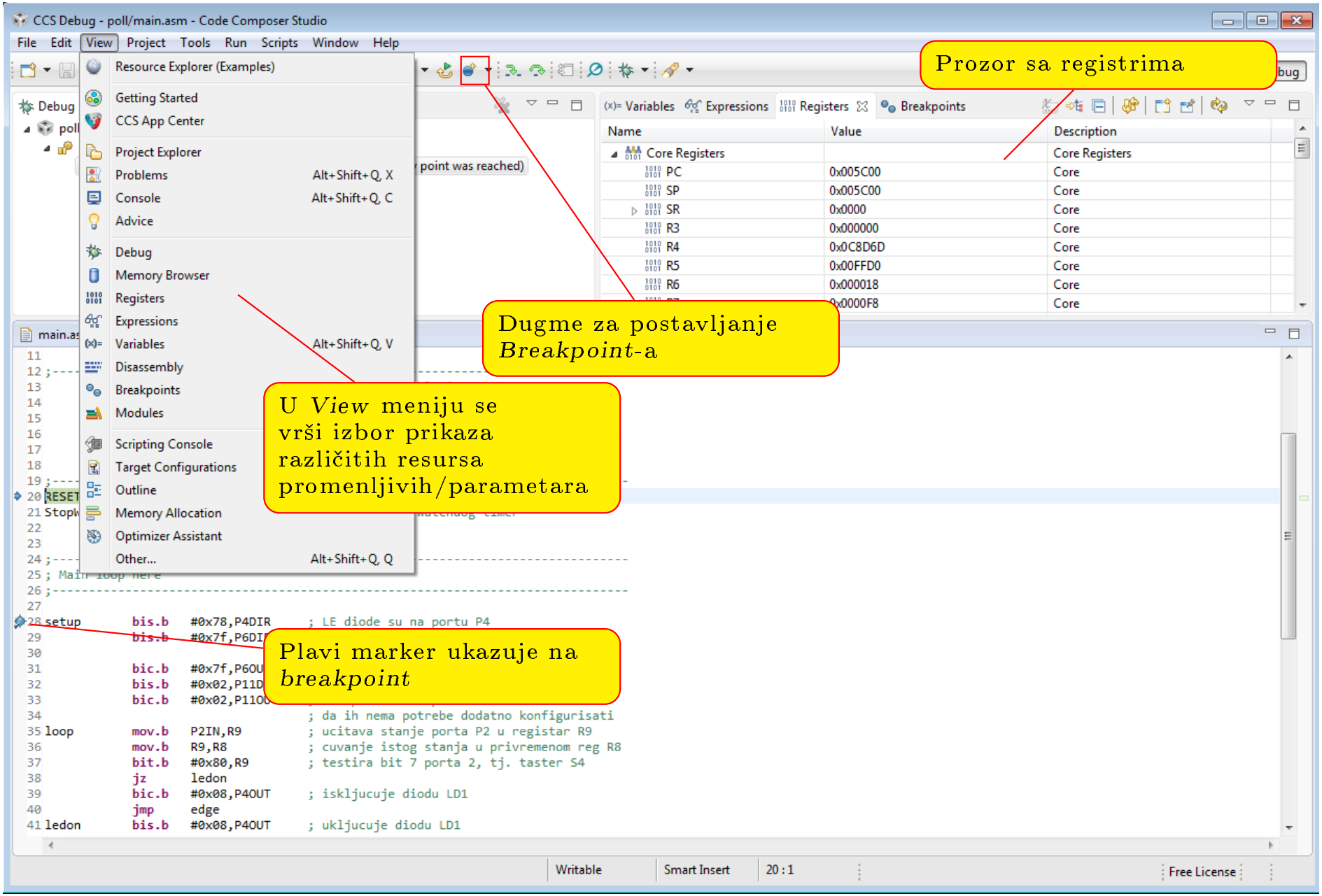


Debagovanje

Debagovanje je proces u kome se uz pomoć hardverskog debagera vrši testiranje aplikacije




CCS Debug - poll/main.asm - Code Composer Studio
File Edit View Project Tools Run Scripts Window Help
Run - klik na ovo dugme izvršava program
Dugme za reset
Step Into - klik na ovo dugme izvršava instrukciju koja je markirana zelenom
Disassembly prozor prikazuje program onako kako je smešten u memoriju
Klikom na ovo dugme prekida se debagovanje
Zeleni marker ukazuje na sledeću instrukciju koja treba da se izvrši
Svakoj liniji originalnog koda koja je prevedena u procesu kompajliranja odgovara jedna linija u Disassembly prozoru
Debug
Registers
Core Registers
main.asm
Writable
Insert
27 : 1
Free License


CCS Debug - poll/main.asm - Code Composer Studio
File Edit View Project Tools Run Scripts Window Help
Resource Explorer (Examples)
Getting Started
CCS App Center
Project Explorer
Problems Alt+Shift+Q, X
Console Alt+Shift+Q, C
Advice
Debug
Memory Browser
Registers
Expressions
Variables Alt+Shift+Q, V
Disassembly
Breakpoints
Modules
Scripting Console
Target Configurations
Outline
Memory Allocation
Optimizer Assistant
Other... Alt+Shift+Q, Q
Prozor sa registrima
Variables Expressions Registers Breakpoints
Name Value Description
Core Registers Core Registers
PC 0x005C00 Core
SP 0x005C00 Core
SR 0x0000 Core
R3 0x000000 Core
R4 0x0C8D6D Core
R5 0x00FFD0 Core
R6 0x000018 Core
Dugme za postavljanje Breakpoint-a
U View meniju se vrši izbor prikaza različitih resursa promenljivih/parametara
Plavi marker ukazuje na breakpoint
setup bis.b #0x78,P4DIR ; LE diode su na portu P4
loop mov.b P2IN,R9 ; ucitava stanje porta P2 u registar R9
mov.b R9,R8 ; cuvanje istog stanja u privremenom reg R8
bit.b #0x80,R9 ; testira bit 7 porta 2, tj. taster S4
jz ledon
bic.b #0x08,P4OUT ; iskljucuje diodu LD1
jmp edge
ledon bis.b #0x08,P4OUT ; ukljucuje diodu LD1
Writable Smart Insert 20 : 1 Free License

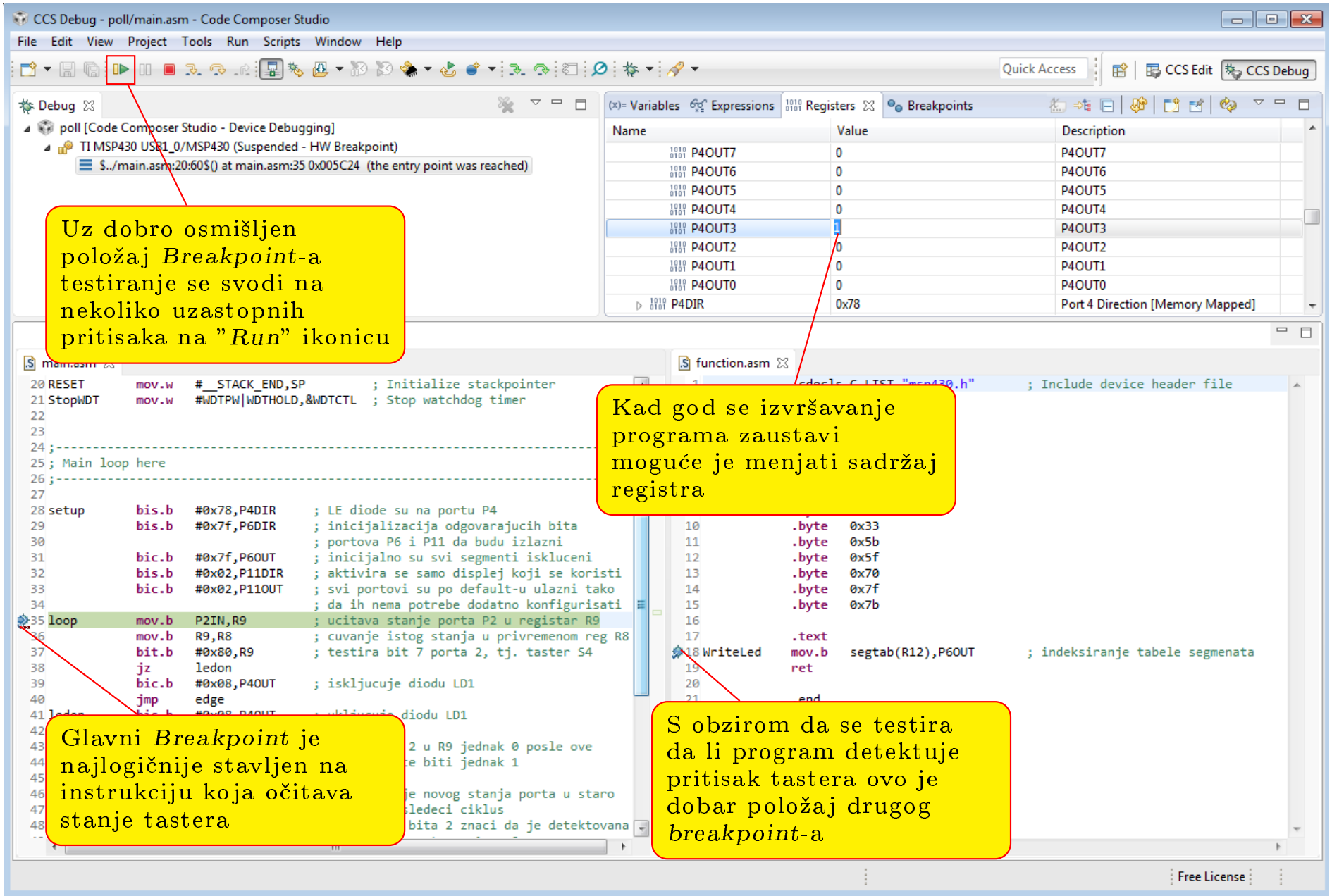
Debagovanje

Najjednostavniji i najpregledniji vid simulacije je *single stepping*

- Problem je što to može dugo da potraje i zamorno je

Mnogo efikasniji vid debagovanja je postavljanjem *Breakpoint*-a na dobra mesta

Generalno, *Breakpoint* se postavlja tamo gde treba simulirati spoljni događaj preko registra (u našem slučaju pritisak tastera) i tamo gde se očekuje reakcija na taj događaj


CCS Debug - poll/main.asm - Code Composer Studio
File Edit View Project Tools Run Scripts Window Help
Quick Access
CCS Edit
CCS Debug
Debug
poll [Code Composer Studio - Device Debugging]
TI MSP430 USB1_0/MSP430 (Suspended - HW Breakpoint)
$../main.asm:20:60$() at main.asm:35 0x005C24 (the entry point was reached)
Variables
Expressions
Registers
Breakpoints
Name
Value
Description
P4OUT7 0 P4OUT7
P4OUT6 0 P4OUT6
P4OUT5 0 P4OUT5
P4OUT4 0 P4OUT4
P4OUT3 1 P4OUT3
P4OUT2 0 P4OUT2
P4OUT1 0 P4OUT1
P4OUT0 0 P4OUT0
P4DIR 0x78 Port 4 Direction [Memory Mapped]
main.asm
function.asm
20 RESET mov.w #__STACK_END,SP ; Initialize stackpointer
21 StopWDT mov.w #WDTPW|WDTHOLD,&WDTCTL ; Stop watchdog timer
24 ;---
25 ; Main loop here
26 ;---
28 setup bis.b #0x78,P4DIR ; LE diode su na portu P4
29 bis.b #0x7f,P6DIR ; inicijalizacija odgovarajucih bita
30 ; portova P6 i P11 da budu izlazni
31 bic.b #0x7f,P6OUT ; inicijalno su svi segmenti iskluceni
32 bis.b #0x02,P11DIR ; aktivira se samo displej koji se koristi
33 bic.b #0x02,P11OUT ; svi portovi su po default-u ulazni tako
34 ; da ih nema potrebe dodatno konfigurisati
35 loop mov.b P2IN,R9 ; ucitava stanje porta P2 u registar R9
36 mov.b R9,R8 ; cuvanje istog stanja u privremenom reg R8
37 bit.b #0x80,R9 ; testira bit 7 porta 2, tj. taster S4
38 jz ledon
39 bic.b #0x08,P4OUT ; iskljucuje diodu LD1
40 jmp edge
10 .byte 0x33
11 .byte 0x5b
12 .byte 0x5f
13 .byte 0x70
14 .byte 0x7f
15 .byte 0x7b
16
17 .text
18 WriteLed mov.b segtab(R12),P6OUT ; indeksiranje tabele segmenata
19 ret
20
21 .end
Uz dobro osmišljen položaj Breakpoint-a testiranje se svodi na nekoliko uzastopnih pritisaka na "Run" ikonicu
Kad god se izvršavanje programa zaustavi moguće je menjati sadržaj registra
Glavni Breakpoint je najlogičnije stavljen na instrukciju koja očitava stanje tastera
S obzirom da se testira da li program detektuje pritisak tastera ovo je dobar položaj drugog breakpoint-a
Free License

Pregled

Povezivanje dioda sa mikrokontrolerom

U primerima koje radimo na ovom kursu koristimo LE Diode. Zajedno sa LE diodama se redno vezuje otpornik koji služi za ograničenje struje diode (videti jednačinu za struju diode)

LE Dioda sa rednim otpornikom vezuje se na pin mikrokontrolera. Pin na koji je vezana dioda mora biti inicijalizovan kao izlazni. Da bi se dioda uključila potrebno je dovesti odgovarajući napon na jedan od krajeva diode.

Električna šema koja ilustruje način vezivanja diode prikazana je na nekom od prethodnih slajdova. Na osnovu prikazane šeme, koji logički nivo uključuje diodu?

1. Zadatak - Blinkanje diode

Zadatak

Napisati program koji sa periodom od 1s uključuje i isključuje diodu na pinu 3 paralelnog porta 4.

1. Zadatak - Blinkanje diode

Rešenje

Ovaj primer ilustruje upotrebu paralelnih portova kada se pinovi na portu koriste kao izlazni. Da bi se pinovi pravilno inicijalizovali i da bi se sistem ponašao na zahtevani način potrebno je uraditi sledeće:

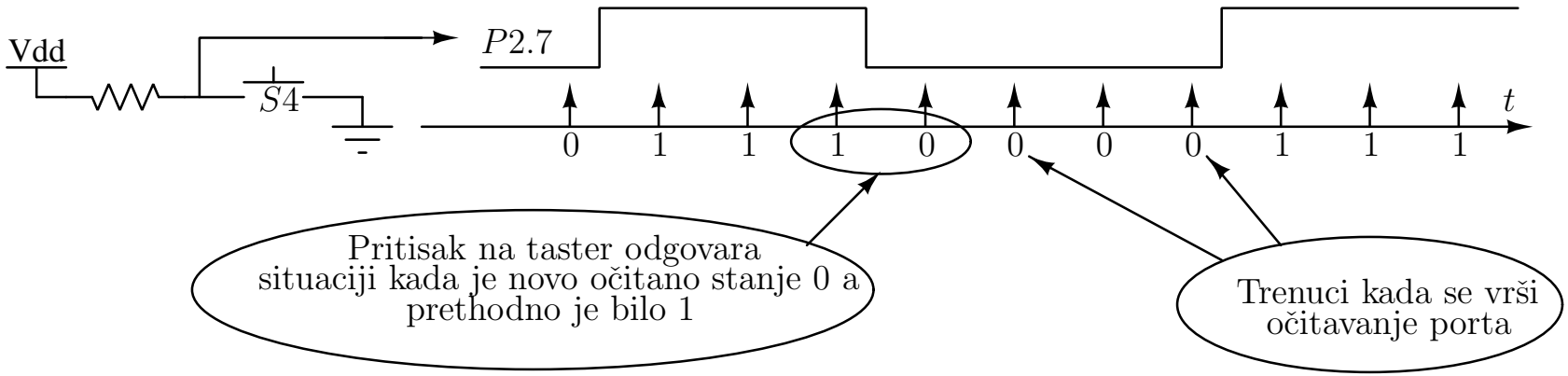
- Isključiti ***watchdog***timer
- Podesiti pin kao izlazni
- Dodeliti pinu inicijalnu vrednost (U ovom primeru je svejedno koja je inicijalna vrednost)
- Menjati stanje diode sa periodom od 0.5s

Kôd

Videti primeri*v2-z1-blink* u materijalima

Detekcija pritiska tastera

Algoritam detekcije pritiska tastera se zasniva na detekciji opadajuće ivice na ulaznom pin-u. Glavni program mikrokontrolera treba periodično da očitava stanje ulaznog pin-a, poredi ga sa vrednosti iz prethodnog očitavanja i u slučaju detekcije prelaska sa 1 na 0 izvrši zahtevanu radnju.



Taster se sa rednim otpornikom povezuje na pin mikrokontrolera. Potrebno je pin mikrokontrolera, na koji je vezan taster, inicijalizovati kao ulazni.

2. Zadatak - Promena stanja diode pritiskom na taster (Detekcija silazne ivice)

Zadatak

Napisati program koji detektuje pritisak tastera S1 i kao rezultat detekcije menja stanje diode LD1.

2. Zadatak - Promena stanja diode pritiskom na taster

Rešenje

Ovaj primer ilustruje upotrebu paralelnih portova kada se pinovi na portu koriste kao ulazni. Pored toga u ovom primeru je ilustrovan i jedan od softverskih načina za detekciju pritiska tastera (detekcija silazne ivice). Da bi se pinovi pravilno inicijalizovali i da bi se sistem ponašao na zahtevani način potrebno je uraditi sledeće:

- Isključiti ***watchdog***timer
- Pin na koji je povezana dioda inicilazovati kao izlazni
- Dioda inicijalno treba da bude isključena
- Inicijalizovati pin na koji je povezan taster kao ulazni pin
- U glavnoj programskoj petlji detektovati pritisak tastera na način objašnjen u nekom od prethodnih slajdova

Kôd

Videti primer *v2-z2-button-toggle* u materijalima

3. Zadatak - Promena stanja 4 diode (Detekcija nivoa signala)

Zadatak

Napisati program koji drži aktivnim sve 4 LE diode za vreme dok je taster pritisnut. Kada taster nije pritisnut diode su isključene.

3. Zadatak - Promena stanja 4 diode (Detekcija nivoa signala)

Rešenje

Ovaj primer ilustruje upotrebu paralelnih portova kada se pinovi na portu koriste kao ulazni. Pored toga ovaj primer ulustruje jedan od načina detekcije niva signala. Zbog načina na koji su tasteri povezani u kolo, nivo koji treba detektovati u ovom primeru je nivo "0".

Da bi se pinovi pravilno inicijalizovali i da bi se sistem ponašao na zahtevani način potrebno je sprovesti niz koraka kao u prethodnom primeru.

Kôd

Videti primeri*v2-z3-button-glow* u materijalima

Prekidi

Jedan od pristupa detekciji pritiska tastera jeste poliranje (stalno ispitivanje da li je došlo do promene stanja tastera). Ovakav pristup nije "efikasan"jer procesor troši dosta vremena na radnju koja se retko dešava.

Zašto mikrokontroler ne reaguje samo kada se pritisne taster?

Mehanizam koji omogućava ovakav pristup naziva se prekid.

Izvori prekida su mnogobrojni (UART, Timer, Paralelni port, DMA ,...)

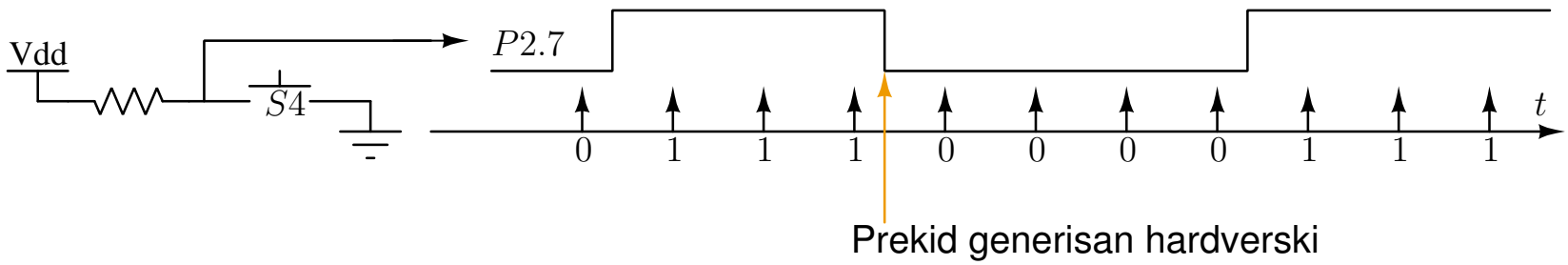
Prekidi portova

Svaki pin na protovima P1 i P2 može da generiše prekid.

Konfiguracija prekida vrši se pomoću registara PxIFG, PxIE i PxIES

Bit u PxIFG registru se setuje kada se desi prekid na odgovarajućem pinu. Pri izlasku iz prekida neophodno je resetovati prethodno setovani bit!

Prekidi koji se generišu promenom stanja ulaznih pinova portova `P1` i `P2`



4. Zadatak - Promena stanja diode

Zadatak

Rešiti 2. zadatak koristeći mehanizam prekida

4. Zadatak - Promena stanja diode

Rešenje

Ovaj primer ilustruje način obrade prekidnog zahteva u prekidnoj rutini. Da bi generisanje prekida bilo moguće potrebno je dozvoliti generisanje prekida i pravilno inicijalizovati pinove koji se koriste. U tu svrhu potrebno je uraditi sledeće:

(U glavnom programu)

- Isključiti ***watchdog***timer
- Pin na koji je povezana dioda inicilazovati kao izlazni i dioda je inicijalno isključena
- Inicijalizovati pin na koji je povezan taster kao ulazni pin i dozvoliti prekid na tom pinu (P2IE)
- Podesiti da se prekid generiše usled silazne ivice(P2IES)

4. Zadatak - Promena stanja diode

Rešenje

Ovaj primer ilustruje način obrade prekidnog zahteva u prekidnoj rutini. Da bi generisanje prekida bilo moguće potrebno je dozvoliti generisanje prekida i pravilno inicijalizovati pinove koji se koriste. U tu svrhu potrebno je uraditi sledeće:

(U prekidnoj rutini)

- Na početku prekidne rutine proveriti izvor prekida (ispitivanje bita P2IFG registra)
- Promeniti stanje diode
- Resetovati ogovarajući bit P2IFG registra

Kôd

Videti primeri*v2-z4-button-toggle-isr* u materijalima

5. Zadatak - Brojač

Zadatak

Potrebno je napisati program koji realizuje funkcionalnost brojača. Vrednost brojača se uvećava za 1 pritiskom na taster S1 dok taster S2 umanjuje vrednost brojača za 1. Opseg vrednosti brojača je 0-255. Kada vrednost brojača izađe iz opsega brojač se "prevrti".

5. Zadatak - Brojač

Rešenje

Ovaj primer ilustruje obradu dva prekidna zahteva koja dolaze od istog izvora prekida. Da bi generisanje prekida bilo moguće potrebno je dozvoliti generisanje prekida i pravilno inicijalizovati pinove koji se koriste. U tu svrhu potrebno je uraditi niz koraka kao u prethodnom primeru!

Kôd

Videti primeri*v2-z5-counter* u materijalima

6. Zadatak - Brojač po modulu 16

Zadatak

Potrebno je napisati program koji realizuje funkcionalnost brojača. Vrednost brojača se uvećava za 1 pritiskom na taster S1 dok taster S2 umanjuje vrednost brojača za 1. Opseg vrednosti brojača je 0-15. Kada vrednost brojača izađe iz opsega brojač se "prevrti". Na sedmosegmentnom displeju (koji se nalazi na razvojnoj ploči) ispisuje se vrednost brojača u heksadecimalnom zapisu.

6. Zadatak - Brojač po modulu 16

Rešenje

Ovaj primer ilustruje obradu dva prekidna zahteva koja dolaze od istog izvora prekida. Da bi generisanje prekida bilo moguće potrebno je dozvoliti generisanje prekida i pravilno inicijalizovati pinove koji se koriste. Pošto se vrednost brojača ispisuje na sedmosegmentnom displeju neophodno je ispravno inicijalizovati i pinove koji su povezani sa ovim displejom.

Kôd

Videti primeri*v2-z6-7seg-counter* u materijalima

Kako ispisati dvocifreni broj na displeju?